Histogram-Based Online Anomaly Detection in Hierarchical Wireless Sensor Networks

Miao Xie, Jiankun Hu*, and Biming Tian
School of Engineering and Information Technology
University of New South Wales at the Australian Defence Force Academy
Canberra, Australia
Email: miao.xie@student.adfa.edu.au j.hu@adfa.edu.au bimingtian@hotmail.com
* Correspondent author

Abstract—Online anomaly detection is critical for protecting wireless sensor networks (WSNs) from cyber-attacks and random faults, which handles the streaming data in real-time. Comparing to other techniques, histogram-based anomaly detection is cheaper in computation, which should be suitable for WSNs. However, performing histogram-based anomaly detection with an online manner in WSNs is not a straightforward issue. Most of the existing histogram-based schemes have to depend on a verification procedure, which costs a great amount of computational overhead as well as communication overhead. Thus, it almost wipes out the advantage of low complexity of histogram-based anomaly detection. This paper introduces a simple estimating approach to detect anomalies with the histogram, which takes account into the distributed manner and online manner at the same time. It also proves the error caused by the new estimate is very small, through a theoretical analysis. Moreover, the optimal parameter will be suggested by minimizing the error. Finally, a set of experiments are implemented with a real WSN dataset, which prove the new scheme is effective and efficient.

Keywords—anomaly detection; distributed computing; histogram; error analysis

I. INTRODUCTION

By analyzing the sensed data or traffic-related features, anomaly detection can protect WSNs from cyber-attacks and random faults [1]. In order to timely handle the streaming data of sensor nodes, anomaly detection is often required to perform with an online manner in WSNs. Online anomaly detection has been addressed by a variety of techniques in WSNs, e.g., rule-based [2], quarter-sphere support vector machine (QS-SVM) [3], fixed-width clustering [4], parametric estimation [5], and kernel density estimator (KDE) [6] [7]. The rule-based schemes are famous for being simple and fast, but they are incapable of defending against unknown cyber-attacks. The QS-SVM-based schemes are constrained by high computational overhead, although the system behaviors can be accurately captured. The fixed-width clustering-based schemes are not adequately reliable, as the results are often affected by the randomness. Without the learning procedures, the parametric estimation-based schemes are computationally cheap. However, these schemes are only good for the applications fitting into the specific statistical assumptions. KDE is a typical technique of nonparametric estimation. In a KDE-based scheme, the system behaviors are represented by an estimated probability density function (PDF), which is obtained through a specific estimator. If a test data occurs with little possibility referring to the estimated PDF, it is regarded as anomalous. Similar to the QS-SVM-based schemes, the KDE-based schemes also suffer from prohibitively expensive computational overhead.

In nonparametric estimation, histogram is also an important technique that estimates PDF with a set of continuous bins. Relying on a histogram, anomaly is detectable similarly to the KDE-based schemes. What is more, histogram is much simpler than KDE [8]. Thus, histogram should be a suited technique to fulfill online anomaly detection in WSNs. For example, Sheng et al proposed a scheme based on equal-width histogram [9], where the base station (BS) and sensor nodes work on the detection cooperatively. Firstly, each sensor node sets its local histogram. Then, the local histograms are aggregated hierarchically with the routing tree, until the BS. After receiving the query message that encloses the global histogram from the BS, each sensor node launches the local detection separately. The detection is realized with the typical k-nearest neighbour (kNN) notion [10]:

A test data $y$ is anomalous in a dataset $X$, if fewer than $k$ data in $X$ are less than or equal to distance $d$ from $y$.

Substantially, the principle is equal to the following problem statement in probability. Suppose that $X$’s PDF is $f(x)$, a test data $y$ is anomalous if $I(y, d) < k$, where

$$I(y, d) = n \int_{y-d}^{y+d} f(x)dx,$$

$d$ is the radius of the detection region, $n$ is the sample size, and $k$ is the detection threshold. Further, it can be rewritten as

$$P(y, d) = \int_{y-d}^{y+d} f(x)dx.$$ 

If $P(y, d) < t$, $y$ is anomalous, where $t = \frac{k}{n}$ is the detection threshold. However, $P(y, d)$ is not available to any $y$, as the histogram only describes the PDF roughly. Instead, there are two properties enabling to identify some anomalies in advance:

- $y$ is normal if residing in a bin of which the integrated probability is greater than or equal to $t$, as this bin is only a sub-region of $[y - d, y + d]$;
- $y$ is anomalous if residing in the middle one of three continuous bins of which the integrated probability is less than $t$, as these bins have fully covered $[y - d, y + d]$.

-
Other than the two cases, the remaining test data should be marked as potential anomalies and then be reported to the BS. When the potential anomalies arrive in the BS, it diffuses a reexamination query message. Each sensor node has to report the summary information about the potential anomalies, which is aggregated hierarchically with the routing tree as well. Eventually, it enables the BS to find out all the true anomalies. When the number of the potential anomalies is large, transmitting the summary information will consume a large amount of communication overhead. In this case, the energy of sensor node will be depleted quickly, which largely reduces the lifetime of the network.

According to $P(y, d)$ and the specific histogram, a straight estimate can be used to detect anomalies, i.e.

$$
\hat{P}(y, d) = \int_{y-d}^{y+d} f(x) dx,
$$

where $f(x)$ is the PDF estimated by the histogram. In practice, $d$ is often set as $h$ immediately, where $h$ is the bin width of histogram. As a result, $\hat{P}(y, d)$ can be rewritten as

$$
\hat{P}(y, h) = \int_{y-h}^{y+h} f(x) dx.
$$

This quantity provides $y$ with a straight estimate regardless of its location. More importantly, this estimate only suffers from a very small error, which is shown in the fifth section. Based on this idea, a new scheme is proposed in this paper.

In addition, it is well-known the sensed data coming from the sensor nodes in close vicinity are often highly correlated [11]. By taking advantage of the spatial correlations, the detection accuracy and robustness will be greatly improved. By the network architecture, it usually divides WSNs into flat and hierarchical [12]. In a flat WSN, the monitor node collects the sensed data from its neighbours for the purpose of anomaly detection [5], which potentially exploits the spatial correlations with the centralized manner. On the contrary, the distributed manner is employed by hierarchical WSNs [3] [6] [7] [9] mostly, where the cluster head (CH, also called as parent node) is responsible for aggregating the global normal profile with the local summaries reported by its common sensor nodes (CNs, also called as child nodes). Obviously, the spatial correlations are retained by the distributed manner as well. However, the distributed manner has two advantages in contrast to the centralized manner. Firstly, it only needs the CH to collect a shorter summary from each CN, which saves communication overhead greatly. Secondly, the computation is spread over the entire network, which will prolong the lifetime of the network. Thus, the distributed manner is preferable for WSNs to perform online anomaly detection. Because flat WSNs are not suitable for the distributed manner, this paper only focuses on hierarchical WSNs.

The online update that enables to capture the latest system behaviours should also be involved in the new scheme, as the system behaviours in the streaming data often vary over time. Because an anomaly is a little possibility event on the specific normal profile, the inaccurate normal profile will result in high false alarm rate. Some schemes have concerned with this issue [6] [7] [9], in which the sliding window is a common solution. In particular, each CN learns the updated summary from the latest training data stored in the sliding window periodically. In order for saving the communication overhead, the CN often delays to process the updated summary until major changes have occurred in the system behaviours. For example, the online update may be only active under certain conditions, or based on a probability.

The rest of this paper is organized as follows. The second section briefly reviews some representative papers in this research field. The preliminaries are given by the third section. Then, the fourth section details the new scheme. In the following section, it provides a theoretical analysis on the error. The sixth section shows the experiments and evaluations. Finally, it concludes this work and identifies a few of research problems for the future study.

II. RELATED WORKS

Da Silva et al designed a rule-based scheme for online anomaly detection [2]. The network only selects a part of sensor nodes as monitor nodes, which take charge of anomaly detection as well as their common functions. The monitor node collects messages by listening in a promiscuous mode. Afterwards, it applies a set of predefined rules to examine the coming message in real-time. Once the number of violations of the rules is beyond a predefined threshold, an alarm is raised. However, finding out a set of rules for all cyber-attacks is impossible. Thus, the detector is powerless to detect unknown cyber-attacks. Secondly, the coverage of the selected monitor nodes cannot be guaranteed, which may make some sensor nodes unattended.

In the QS-SVM-based scheme [3], a hyper quarter-sphere classifies the sensed data in the feature space (Hilbert space), where the network works in a hierarchical topology. At first, each CN obtains the local classifier (i.e. the optimal radius of the local hyper quarter-sphere) from its sensed data. Afterwards, the CH combines all the local classifiers into a global classifier. Each CN performs anomaly detection in real-time, based on both the local classifier and global classifier. Overall, the performance is acceptable for this scheme, but the sensor node may be not affordable to the expensive computational overhead. The fixed-width clustering-based scheme [4] equips each sensor node with a detector individually. The sensed data are divided into a set of clusters in a sensor node. The clusters which are sparse below a threshold will be marked as anomalous clusters, otherwise normal clusters. When receiving a new sensed data, it is being tested with these clusters immediately. If the test data is out of any clusters or falling into an anomalous cluster, it is anomalous. This scheme does not take advantage of the correlations existed in the spatially proximal sensed data, which may degrade the detection accuracy. On the other hand, fixed-width clustering is largely affected by the randomness, thus not very reliable.

Liu et al proposed a parametric estimation-based scheme [5], in which it assumes the sensed data and traffic-related features are statistically subject to normal distribution. Anomaly detection is being operated in a flat WSN, where each sensor node monitors its one-hop neighbours. If a test data is unusually larger than the expected value, it is regarded as anomalous. This scheme is simple in computation; however,
each sensor node has to act as the monitor node and be the object monitored by the others simultaneously. A large amount of unnecessary resource overhead will be consumed. What is worse, the detector will be disabled when the real distribution largely deviates from normal distribution. Palpanas et al [6] and Subramaniam et al [7] developed the KDE-based schemes for univariate data and multivariate data respectively. In a hierarchical WSN, each CN estimates its local distribution, which is then reported to the CH. The CH combines all the local distributions into a global one. Each CN timely examines the test data with a distance-metric or a multi-granular metric, referring to its local distribution. All the local anomalies will be sent to the CH for verifying, where the global distribution is made use. The normal data in a CN will be transmitted to the CH according to a predefined probability, so as to update the normal profile periodically. Similar to the QS-SVM based scheme, high computational overhead is the main bottleneck for the KDE-based schemes.

Base on histogram, a typical example is the scheme proposed by Beigi et al [8]. Firstly, a histogram is settled with the historical data, called as baseline. The observed data (stream) is stored in a size-flexible sliding window, which enables to detect the data segment at multiple temporal scales simultaneously. A distance vector measures the difference between the observed data and the baseline. By the specific distance metric and predefined threshold, anomaly is detectable. Furthermore, the detection accuracy is guaranteed through minimizing the asymptotic mean integrated squared error (AMISE). In order to capture the system behaviours accurately, this scheme has to be dependent on a large number of historical data. The distance computation and stream processing are also computationally expensive. Thus, it is an ideal scheme being operated in the BS, but not suited for in-network computation.

Taking advantage of the straight estimate based on a histogram, online anomaly detection will run smoothly, as it doesn’t need the verification procedure any more. Accordingly, a new scheme is proposed in the following, which satisfies:

- identifying anomalies without repeated verification
- operating in an online manner
- operating in a distributed manner

To the best of our knowledge, this is the original effort made to consider online anomaly detection with such an idea in hierarchical WSNs.

III. PRELIMINARIES

A. Histogram

A histogram is composed of a set of continuous bins \( B_k \) \((k = 1, 2 \cdots)\), normally with fixed width \( h \) [13]. The histogram represents the estimated PDF by

\[
\hat{f}(x) = \frac{V_k}{n_h}, \quad k = 1, 2 \cdots
\]

where \( V_k \) is the number of data in \( B_k \). Note that \( V_k \) is actually subject to binomial distribution, i.e.

\[
V_k \sim B(n, p_k)
\]

where \( p_k = \int f(x)dx \). Further, there are

\[
\begin{align*}
E(V_k) &= np_k \\
\text{var}(V_k) &= np_k(1 - p_k)
\end{align*}
\]

B. Mean Squared Error

Mean squared error (MSE) is a typical evaluation criterion for an estimate in statistical community, which is defined as

\[
E(\bar{P} - P)^2 = \text{var}(\bar{P}) + \text{bias}^2,
\]

where \( P \) is a variable, \( \bar{P} \) is an estimate, and

\[
\text{bias} = E(\bar{P}) - P.
\]

In general, a smaller MSE corresponds to a better estimate. MSE measures the error in the pointwise sense. In the global sense, the quantity mean integrated squared error (MISE) has to be found out, which is defined as

\[
\int E(\bar{P} - P)^2 = \int \text{var}(\bar{P}) + \text{bias}^2.
\]

Similarly, a small MISE suggests the estimate is close to the real value on average.

C. Network Model

The network model is defined as follows. In order to simplify the problem, the network is assumed as a two-layer hierarchical structure. If there are multiple layers, it only needs to iteratively deploy the scheme in each layer, for example [9]. Particularly, one layer is between the clusters and BS, while the other layer is between the CH and CNs. The network is composed of many clusters, which will not interfere mutually. In each of the clusters, a node will be elected as the CH in turn that takes charge of data fusion and communication with the BS, while the rest of nodes are called as CNs. When the detector is running, the network topology is assumed to maintain static. It also assumes that all the sensor nodes are time synchronized. Figure 1 gives an example illustratively.
IV. ONLINE ANOMALY DETECTION BASED ON HISTOGRAM

The scheme only needs to be considered in a cluster separately. Given a cluster comprised of m CNs, the ith CN conducts the local dataset $X_i$ by a sliding window of size n, where $i = 1 \cdots m$. In this paper, the scheme only focuses on univariate data, which may extend to multivariate later. In theory, both the sensed data and traffic-related features are suitable data sources for anomaly detection. However, it has not been confirmed yet the spatial correlations also exist in the traffic-related features. As a result, it employs the sensed data as the data source uniquely. In addition, it assumes the sensed data are standardized, so as to facilitate the theoretical analysis.

The scheme consists of three phases: training, test, and update. During the training phase, the CH aggregates the local histograms reported by the CNs as a global histogram. Then, each CN individually marks out those normal and anomalous bins according to the histogram, before it continues to the test phase. During the test phase, each CN performs detection with either or both of the local and global estimates in real-time, each of which contains a histogram and a mark array. Finally, it returns the test data with a probability, which will be used for capturing the newest histogram during the update phase.

A. Training Phase

When the detector starts up, each CN collects the sensed data until its local sliding window is full. As the local histograms will be aggregated globally, each of them should be settled with a same baseline. Normally, the CH needs to know the global value range of the sensed data. The global value range is given by the maximal and minimal values of the sensed data throughout all the CNs. Afterwards, each CN sets the local histogram with the global value range, which counts the bins from the minimal value to the maximal value continuously. Such a method gives a fixed boundary to the histogram, leading to two issues. On the one hand, it cannot accurately detect a test data being out of the fixed boundary, when its detection region has overlap with the histogram. On the other hand, the global value range has to be updated correspondingly during the update phase, which costs added resource overhead. If it counts the bins from the origin of the coordinate (zero point) to infinity, there is no longer a fixed boundary. At this time, the histogram can be settled boundlessly, which adapts to any situation. Besides, the resource overhead spent on settling the global value range is exempt. In particular, a data $x$ belongs to $B_k$, if

$$\begin{align*}
\left\{ \begin{array}{l}
\frac{x}{h} = k, x \geq 0 \\
\frac{x}{h} - 1 = k, x < 0
\end{array} \right.
\end{align*}$$

Each CN examines its local training data sequentially with the expression above, which produces a sequence of size n that records the bin positions. This sequence contains redundant information as some bin positions may appear multiple times. Thus, it is compressed into a two-dimensional array, where one records the bin position while the other one records the corresponding frequency of occurrence. Before counting the frequency of occurrence, it should sort the sequence in ascending order, such that the compression can be completed by only one time scan. The histogram is described by the two-dimensional array in fact. After the CH collects all the local arrays from the CNs, the global histogram is produced as follows. The local arrays are simply put together first of all. Then, it is sorted in ascending order in terms of the bin position. The resulted global two-dimensional array can be compressed similarly to that of the local sequence, by which the global normal profile is produced eventually. Finally, the CH broadcasts the normal profile to the MNs.

B. Test Phase

Online anomaly detection begins with the equation

$$\hat{P}(y, h) = \int_{y-h}^{y+h} f(x)dx.$$  

Given a test data $y$ that falls into the bin $B_{k1}$, there are

$$\begin{align*}
\left\{ \begin{array}{l}
a = k_2 h - y \\
b = y - k_1 h
\end{array} \right.,
\end{align*}$$

where $k_2 = k_1 + 1$. Figure 2 exemplifies this scenario. Thus, there are

$$\begin{align*}
\hat{P}(y, h) &= \int_{y-h}^{y+h} f(x)dx + \int_{k_1}^{k_2} f(x)dx + \int_{k_2}^{y+h} f(x)dx \\
\hat{P}(y, h) &= \frac{ak}{h} + \frac{b}{n} + \frac{V_{k1}b}{n}.
\end{align*}$$

If $\hat{P}(y, h) < t$, $y$ is anomalous, where $t$ is the predefined threshold.

Similar to [9], the two deterministic properties also make sense in this case, which enable to detect a significant portion of the test data without knowing $\hat{P}$. Therefore, each CN marks the bins as normal or anomalous, in accordance with the following rules:

- $B_k$ is a normal bin, if $\frac{V_k}{n} \geq t$;
- $B_k$ is an anomalous bin, if $\frac{V_{k-1} + V_k + V_{k+1}}{n} < t$;
- Otherwise, $B_k$ is not marked.

It only needs to detect those test data falling into the unknown bins with $\hat{P}(y, h)$, such that a large amount of computational overhead is saved.

There are two sets of estimates in each CN at the same time, which are referred to as local and global respectively. Each set consists of $f$ and the array that records the marks of bins. The CN makes use of either or both of them optionally. When the local distribution is quite close to the global distribution,
satisfactory performance can be achieved by any set of them alone. Otherwise, the test data should be detected with both of them, to guarantee the detection accuracy and robustness. In this case, y is regarded as normal only if it passes the examinations by both the local and global estimates. The duplicated detection is not compulsory, as it doubles the computational overhead. Note that the threshold t may be specified for the two sets of estimates differently.

C. Update Phase

The test data will be used for the online update according to a probability, which can reduce the communication overhead as well as the computational overhead. Therefore, the training phase will restart periodically. Specifically, the test data is kept with the probability $p_a$. This is equal to sample the proportion $p_u$ from the next data stream randomly. If the PDF changes smoothly, it decreases the update frequency by using a smaller $p_u$; otherwise, $p_u$ increases. The selected test data is pushed into the sliding window, taking the place of the oldest data. When a certain proportion of the sliding window has been updated (e.g. $\frac{1}{2}$), the CN launches a new round of training. The next procedure is almost same with what the training phase does. The CN reports its updated local histogram to the CH, by which the CH resolves the updated global histogram. Finally, the CH sends the updated global histogram to the CNs. Such a method is useful for saving the resource overhead. For example, the CN only needs to update once after 400 times of detection have been done, if the proportion threshold is set as $\frac{1}{3}$.

D. Completed Scheme

The CH and CNs cooperatively perform the scheme, where the CH takes part in the training phase and update phase and the CNs attend all the phases. The figure Completed Scheme describes the completed scheme.

V. ERROR ANALYSIS

By analyzing $P$’s MISE, two useful results will be found: 1) how much error will be generated in theory; and 2) how to minimize the error. $P$’s MISE is derived as follows.

With the original form of $P$, $P(x, d) = \int_0^{x+d} \bar{f}(x)dx$, the variance is analyzed at first. When $d = h$, $P$ normally overlays over three adjacent bins, as shown in Figure 2. Given $\frac{\sum q_k}{nh} = \bar{f}_k$, there is

$$\text{var}(P) = \text{var}\left(\int_{x-h}^{x+d} \bar{f}(x)dx\right)$$

$$= \text{var}\left(\int_{x-h}^{0} \bar{f}_0dx + \int_{0}^{h} \bar{f}_1dx + \int_{h}^{x+d} \bar{f}_2dx\right)$$

$$= a^2\text{var}(\bar{f}_0) + d^2\text{var}(\bar{f}_1) + b^2\text{var}(\bar{f}_2) +
2(ad\text{cov}(\bar{f}_0, \bar{f}_1) + ab\text{cov}(\bar{f}_0, \bar{f}_2) + bd\text{cov}(\bar{f}_1, \bar{f}_2))$$

where $a + b = d$. Note that

$$\int \text{var}(\bar{f}_k)dx = \sum_k \text{var}(\bar{f}_k)dx = \sum_k p_k(1-p_k)$$

where $k_i = 1, 2 \cdots$. Based on mean value theorem (MVT), and Riemannian integral approximation.

![Completed Scheme](image)

1. collect the training dataset $X_i$
2. examine $x \in X_i$ with
   $\left\{\begin{array}{ll}
   \text{SEQ}[i] = \lfloor \frac{x}{nh} \rfloor, & x \geq 0 \\
   \text{SEQ}[i] = \lfloor \frac{x}{nh} \rfloor - 1, & x < 0
   \end{array}\right.$
3. sort(SEQ, ascending)
4. compress SEQ into (LH.pos, LH.num)
5. LH $\rightarrow$ CH, GH $\leftarrow$ CH
6. mark LH and GH by LM and GM
7. test data $y$ arrives in
8. detect $y$ with LM, GM, if passed continue to 10
9. detect $y$ with $P(y, h)$, by LH and CH
10. retain the test data with probability $P_u$
11. repeat 7-10 until meets update condition
12. continue to 2

in the CH
1. LHs/updated LH $\leftarrow$ CNs
2. combine LHs into GHT
3. sort(GHT, GHT.pos, ascending)
4. compress GHT into (GH.pos, GH.num)
5. GH $\rightarrow$ CNs
6. when receiving update request, continue to 1

$$\sum p_k = 1, \sum p_k^2 = \sum_{k_i} f(\xi_k)^2h^2 = h \int f(x)^2dx + O(1),$$

where $\xi_k \in B_{k_i}$. Thus,

$$\int \text{var}(\bar{f}_{k_i})dx = \int \text{var}(\bar{f}(x))dx = \frac{1}{nh} \frac{R(f)}{n} + O(n^{-1}),$$

where $R(f) = \int f(x)^2dx$. The same result can be found for $B_{k_0}$, $B_{k_1}$, and $B_{k_2}$. On the other hand,

$$\text{cov}(\bar{f}_{k_i}, \bar{f}_{k_j}) = \frac{1}{nh^2} \frac{n(p_{kij} - p_{k_i}p_{k_j})}{n},$$

where $p_{kij}$ is the probability that $V_{k_i}$ and $V_{k_j}$ equal to 1 simultaneously when $n = 1$. However, $V_{k_i}$ and $V_{k_j}$ are impossible to be 1 at the same time when $n = 1$. Thus, $p_{kij} = 0$ and $\text{cov}(\bar{f}_{k_i}, \bar{f}_{k_j}) = \frac{n(p_{kij} - p_{k_i}p_{k_j})}{nh^2}$. This covariance can be equal to $-\frac{f(\xi_{kij})^2}{n}$ approximately [13], where $\xi_{kij}$ is the center point between $B_{k_i}$ and $B_{k_j}$. Using Riemannian integral approximation,

$$\int \text{cov}(\bar{f}_{k_i}, \bar{f}_{k_j})dx = \int -\frac{f(\xi_{kij})^2}{n}dx = \frac{1}{n} \sum h(f(\xi)^2 = \frac{R(f)}{n} + O(n^{-1}).$$

Substituting $\int \text{cov}(\bar{f}_{k_i}, \bar{f}_{k_j})dx$ and $\int \text{var}(\bar{f}_{k_i})dx$, the integrated variance (IV) is

$$\int \text{var}(P) = \frac{(a^2+b^2+d^2)(\frac{1}{nh})^2}{n^2} \frac{R(f)}{n} - 2(ad + ab + bd)\frac{R(f)}{n} = \frac{(a^2+b^2+d^2)(\frac{1}{nh})^2}{n^2} - 4d^2\frac{R(f)}{n}.$$}

Given $z_1 = a^2+b^2+d^2$, and $z_2 = 4d^2$, IV $= z_1 \frac{1}{nh} - z_2 \frac{R(f)}{n}$.

Subsequently, the bias is analyzed, which is

$$\text{bias}(P) = \text{bias}(\int_{x-d}^{x+d} f(x)dx).$$

According to MVT,

$$\text{bias}(P) = 2d\text{bias}(\bar{f}(\xi)),$$

where $\xi \in [x-d, x+d]$. It is known that [13]
The integrated squared bias (ISB) should be

\[ ISB = \int \text{bias}^2(\hat{P}) \text{d}x = \int 4d^2 \text{bias}^2(\hat{P}(x)) \text{d}x \approx 4d^2 \int \text{bias}^2(\hat{P}(x)) \text{d}x . \]

Due to the two main terms of IV and ISB, \( P \)'s asymptotic MISE (AMISE) is

\[ \text{AMISE}(h) = \frac{\pi^4}{96} + \frac{\pi^2}{3} h^2 R(f'). \]

Based on this equation, the following theorems will be easily drawn out.

**Theorem 1** The optimized \( h \) is \( h^* = \left( \frac{6}{2} \frac{1}{\text{var}(f')} \right)^{\frac{1}{2}}. \)

**Proof**

\[
\begin{align*}
\frac{\text{d(AMISE(h))}}{\text{d}h} &= 0 \\
\rightarrow \frac{\text{d}h}{\text{d}h} - 2 + \frac{2\pi}{3} \sqrt{\text{var}(f')} & h = 0 \\
\rightarrow h^* = \frac{1}{\sqrt{2 \frac{1}{\text{var}(f')}}} & \frac{\pi^2}{3} h^2 R(f') \Rightarrow h^* = \left( \frac{6}{2} \frac{1}{\text{var}(f')} \right)^{\frac{1}{2}}.
\end{align*}
\]

Due to \( a + b = d \) and \( d = h, \ \frac{\pi^2}{3} \) should belong to \( \left[ \frac{2}{\sqrt{2}}, 1 \right] \) in terms of \( z_1 \in \left[ \frac{\pi^2}{6} h^2, 2h^2 \right] \) and \( z_2 = 4h^2 \). Thus, \( h^* \) belongs to \( \left[ \frac{6}{2} \frac{1}{\text{var}(f')} \right]^{\frac{1}{2}}, \left( \frac{3}{\sqrt{2 \text{var}(f')}} \right)^{\frac{1}{2}} \). Taking \( h^* \) into AMISE(h), AMISE(h) will be worked out, i.e.

**Theorem 2** The asymptotically minimized error varies in \( \left[ \frac{1.4}{n} + 3.93(nR(f'))^{-\frac{1}{2}}, \frac{1}{n} + 5.77(nR(f'))^{-\frac{1}{2}} \right], \) in terms of \( z_1 \in \left[ \frac{\pi^2}{6} h^2, 2h^2 \right] \) and \( z_2 = 4h^2 \).

Using E1 (see the following section) as the example, the acceptable performance is acquired by \( h \in [0.2, 0.4] \), where \( n = 1500 \). As the distribution is unknown in practice, it may as well assume \( f \sim N(0, 1) \) where \( N \) is Gaussian distribution, and \( R(f') = 0.141 \) [13]. According to Theorem 1, the optimal \( h \) should be varying in \( [0.22, 0.24] \), which coincides with the experimental result. On the other hand, AMISE(h) will be up to \( 5.9 \times 10^{-3} \), according to Theorem 2. Compared to the threshold \( t \), \( t \in [0.05, 0.15] \), such an error is pretty small. Accordingly, the error will not degrade the performance remarkably.
C. Evaluation

In E1, valid value range of \( h \) is \([0.2, 0.4]\), where the detection accuracy (1-FNR) stays above 80% generally and the FPR is below 30%. In E2.1, acceptable performance can be achieved by a larger value range of \( h \), \([0.25, 0.5]\). In that case, the detection accuracy is higher than 85% in general, while FPR is never above 20%. The degradation in the performance occurs in E2.2, where the detection accuracy varies around 70% when \( h \) belongs to \([0.01, 0.2]\) and FPR is higher than 30%. By observing Figure 4 and Figure 7(A), it can be found that such degradation is not caused by the probability-based online update. Figure 7(A) shows the sampling is completely able to capture the dynamics of system behaviours. In fact, there is a sharp decrease after 500th data in node 8, which is shown in Figure 4. Tackling such abrupt changes in streaming data is still a common issue that all the existing online anomaly detection techniques have to face. As far as the online update, it will not remarkably impact on the performance unless meeting the abrupt change, because the dynamics has been captured by the sampling.

The proportion of test data being detected by the marked bins is not regularly changed. However, at worst there is also about one third of test data detectable with the marked bins (\( h = [0.01, 0.2] \), E2.2). In E2.1, even only 5% test data are required to know \( P_2 \), when \( h = [0.25, 0.5] \). Thus, a large amount of computational overhead will be saved with this method.

VII. Conclusion

The development of online anomaly detection for WSNs is a hot topic in recent years. The guarantee on the generality and reliability at the same time is very challenging due to the intrinsic shortages of WSNs. Histogram should have been a good technique for anomaly detection, owing to its lower complexity. However, existing histogram-based schemes have to depend on an additional verification procedure as they are unable to accurately measure the test data with the histogram alone. In the sense of probability, a straight estimate is proposed to identify anomalies with the same concept, where the verification procedure is exempt. Moreover, this estimate only takes a very small error in theory. In consequence, a novel anomaly detection scheme is proposed in this paper, which
TABLE II
EXPERIMENTAL PARAMETERS

<table>
<thead>
<tr>
<th>Feature</th>
<th>(m)</th>
<th>(n)</th>
<th>training data</th>
<th>normal data</th>
<th>anomalous data</th>
<th>(h)</th>
<th>(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Temperature</td>
<td>3</td>
<td>500</td>
<td>1500</td>
<td>100</td>
<td>100</td>
<td>0.01, 0.15</td>
</tr>
<tr>
<td>E2.1</td>
<td>Humidity</td>
<td>3</td>
<td>300</td>
<td>900</td>
<td>100</td>
<td>100</td>
<td>0.01, 0.15</td>
</tr>
<tr>
<td>E2.2</td>
<td>Humidity</td>
<td>3</td>
<td>300/600</td>
<td>900/1800</td>
<td>100</td>
<td>100</td>
<td>0.01, 0.15</td>
</tr>
</tbody>
</table>

Fig. 5. Results of E1, optimal \(h \in [0.2, 0.4], t = 0.12\)

Fig. 6. Results of E2.1, optimal \(h \in [0.25, 0.5], t = 0.05\)

takes account into both the distributed manner and online operation.

In terms of the particular error analysis, it proves not only the error is within an acceptable range, but also potentially suggests how to select the optimized parameter. The experiments demonstrate the suggested parameter is very effective. The future study will focus on several issues. Firstly, this scheme should be extended to support multivariate data, which would improve the generality and detection performance further. Secondly, it is possible to design an automated parameter selection algorithm. \(R(f')\) is the unique variable to determine the optimized \(h\). Although \(R(f')\) is unknown in general, it remains possible to estimate it with the dataset. Then, the optimized \(h\) can be found out in advance. Finally, the online update should be provided with the ability to perceive the unexpected change of normal behaviors, and accordingly
Fig. 7. Results of E2.2, optimal $h \in [0.01, 0.2]$, optimal $t = 0.05$

optimizes its strategies in real-time.

In the future work, we are also interested to apply these anomaly detection techniques to address the high false alarm rate problem in existing anomaly intrusion detection systems [15] [16] [17] [18].

REFERENCES


